1   METHOD AND APPARATUS FOR TRANSMITTING VIDEO SIGNALS

2

3   CROSS REFERENCE TO RELATED APPLICATIONS

4

5   This application is a continuation-in-part of application

6   serial number 10/233,299, filed August 28, 2002.

7

8   TECHNICAL FIELD OF THE INVENTION

9   The present invention is directed generally to the field

10 of transmitting video signals. Specifically, the present

11 invention relates to a method and apparatus for effectively

12 multicasting video signals by digitizing and compressing

13 received video signals such that they may be transmitted with

14 other digitized signals directly or via a network, such as the

15 Internet, to a multitude of computers and/or independent video

16 display devices, whereupon the video signals are decompressed

17 and displayed.

18

19 BACKGROUND OF THE INVENTION

20   A few types of video transmission systems are commonly

21 used today including multicast, broadcast, and point-to-point

22 video transmission systems. Multicast and broadcast systems

23 both transmit video signals from one source to multiple

1 recipients simultaneously. However, broadcast systems

2 transmit to everyone, whereas multicast systems transmit only

3 to a specific group. In contrast, a point-to-point video

4 transmission system transmits video from one source to one

5 recipient at any given time. A point-to-point video

6 transmission system may have multiple video signal recipients,

7 but this type of system requires multiple video signal

8 transmissions (i.e., one for each recipient).

9 Video transmission systems are widely used for many

10 purposes. One such purpose is to allow individuals at one or

11 more remote sites to visually and/or audibly participate in a

12 local presentation without physical relocation. In this

13 scenario, the video transmission system typically includes a

14 video source, such as a video camera or computer, at the

15 location at which the live presentation occurs. Also, each

16 participant location is equipped with a video display unit for

17 viewing the presentation. The video signals may then be

18 transmitted from the local presentation site to each

19 participant location using any one of a variety of

20 communication mediums.

21 One such communication medium is a dedicated line, such

22 as a telephone or cable line. These lines may be provided in

23 a variety of forms including a twisted pair telephone line, a

1 Digital Subscriber Line ("DSL"), an Integrated Services

2 Digital Network ("ISDN") line, a T1 line, a cable line, etc.

3   When implementing a video transmission that operates with

4 a dedicated line, the user incurs many costs. First, the

5 system user typically pays an ongoing fee for continued use of

6 the dedicated line. Second, the system user must purchase

7 video transmission system equipment that is compatible with

8 the chosen type of dedicated line. Third, the system user

9 must pay for installation, wiring, and setup of the custom

10 equipment. In addition to the relatively high cost of such

11 systems, the scalability of these systems is limited to those

12 locations to which the dedicated line is connected. In many

13 instances, this may include only two or three locations. For

14 major corporations having hundreds of worldwide offices,

15 employees at non-connected locations will still need to travel

16 to a connected location. Alternatively, dedicated telephone

17 lines may be purchased such that every location is connected

18 to the dedicated line, however, this typically results in

19 tremendous installation and maintenance costs. Consequently,

20 choosing a dedicated line as the communication medium for

21 implementation of a video transmission system having many

22 participant locations is typically cost prohibitive. Also,

23 such systems are typically not compatible with an independent

1 video source that has not been modified or designed for

2 compatibility with the custom equipment.

3   An alternative communication medium is an existing

4 computer network such as a Local Area Network ("LAN") or Wide

5 Area Network ("WAN"). Such systems are typically server-

6 based. That is, the same server that controls the flow of all

7 network information also controls the flow of the multicast

8 video signals. Such systems are traditionally more economical

9 than dedicated line systems, as they do not require a

10 dedicated infrastructure. Rather, such systems utilize

11 existing computer network infrastructures, thereby minimizing

12 wiring costs. However, since these systems are server-based,

13 they typically only allow participants to receive multicast

14 video signals via a computer connected to the computer

15 network. In other words, each participant location must be

16 equipped with a video monitor and a networked computer.

17 Furthermore, the video source, such as the computer used to

18 provide the presentation, must also be connected to the

19 network to allow the server to receive the video signals.

20 Additionally, some networked systems require the video source

21 to execute application specific software to allow its video

22 signals to be transmitted via the network. Therefore,

23 although this type of video transmission system may be more

1 economical, it is not compatible with independent, non-

2 networked video sources.

3   Yet another alternative communication medium is the

4 Internet. Such systems typically require the system user to

5 purchase a subscription from a service provider. This

6 subscription allows the system user to utilize the service

7 provider's server-based infrastructure to manage

8 communications from the system user's computer or video camera

9 to the intended recipients via the Internet. The service

10 provider's infrastructure receives the video signals generated

11 by the system user via the Internet, whereupon the signals are

12 processed and transmitted in a point-to-point manner to each

13 of the designated recipients also via the Internet. Such

14 service providers typically couple the video transmission with

15 a telephone conference call to allow interactive visual and

16 audible presentations. For example, a system user typically

17 schedules a presentation with a service provider to occur at a

18 specific date and time. Thereafter, this information is sent

19 to all intended participants along with password data and

20 information regarding how to connect to the presentation. At

21 the scheduled time, the participants connect to the

22 presentation visually through their computer by logging on to

23 the service provider's web page and entering the required

1 password and/or session information. Similarly, the

2 participants connect to the presentation audibly by dialing a

3 designated telephone number and entering the required password

4 and session data. Such a service typically requires continual

5 payment of a subscription fee and purchase of equipment setup

6 prior to system use. Also, these systems are typically

7 server-based and the service is limited to those features

8 offered by the service provider. Furthermore, these service

9 providers do not offer recording of presentations for later

10 review and/or playback.

11   Many of the aforementioned communication mediums are

12 digital mediums. When a digital medium is utilized to

13 transmit analog video signals, these analog signals must be

14 converted to a digital format prior to transmission via the

15 digital medium. These digital signals, in uncompressed form,

16 require a communication medium having a large bandwidth (i.e.,

17 transmission capacity) if these signals are to be transmitted

18 in near real-time. Generally, even high-speed connections

19 such as cable and DSL are incapable of accommodating such

20 bandwidth requirements. Furthermore, a majority of home users

21 still connect to the Internet via a traditional twisted pair

22 telephone line having even less bandwidth than their high-

23 speed alternatives. Therefore, if a video transmission system

1 is to be compatible with any type of communication medium, the

2 converted digital video signals must be compressed prior to

3 transmission and decompressed after transmission.

4   The video transmission system of the present invention

5 uses the compression algorithm disclosed in co-pending

6 application serial no. 10/233,299, which is incorporated

7 herein by reference, to reduce and compress the digital data

8 that must be transmitted to the remote computers and/or video

9 display devices. Generally, video signals generated by a

10 personal computer have large amounts of both spatial and

11 interframe redundancies. For example, in a near idle personal

12 computer, the only change between successive frames of video

13 might be the blinking of a cursor. Even as a user types a

14 document, a majority of the screen does not change over a

15 period of time. Hence, the compression algorithm used by the

16 present invention takes advantage of these redundancies, both

17 between successive frames of video and within each individual

18 frame, to reduce the amount of digital video signal data that

19 is transmitted to the remote computers and/or video display

20 devices. Reducing the amount of digital data transmitted over

21 the communication medium decreases communication time and

22 decreases the required bandwidth.

1   Most forms of video compression known in the art require

2 complicated calculations. For example, Moving Pictures

3 Experts Group ("MPEG") video compression algorithms use the

4 discrete cosine transform as part of its algorithm. Also, the

5 MPEG standard relies on the recognition of "motion" between

6 frames, which requires calculation of motion vectors that

7 describe how portions of the video image have changed over a

8 period of time. Since these algorithms are calculation

9 intensive, they either require relatively expensive hardware

10 that performs such calculations quickly or extended

11 transmission times that allow sufficient time for slower

12 hardware to complete the calculations.

13   In addition to complexity, many existing video

14 compression techniques are lossy (i.e., they do not transmit

15 all of the video signal information in order to reduce the

16 required bandwidth). Typically, such lossy techniques either

17 reduce the detail of a video image or reduce the number of

18 colors. Although reducing the number of colors could be part

19 of an adequate compression solution for some video

20 transmission system applications, in many other applications,

21 such a result defeats the intended purposes of the video

22 transmission system.

1   Several patents are directed to the field of video

2   transmission systems. For example, Zhu *et al.* U.S. Patent No.

3   6,567,813 ("Zhu"), assigned on its face to WebEx

4   Communications, Inc., discloses a distributed, collaborative

5   computer system that includes a plurality of server computers

6   connected via a high speed link. The Zhu system allows client

7   computers to connect to any available server computer to start

8   or join a conference hosted on either the server computer to

9   which the client computer is connected or to any other server

10   in the system. This system is server-based, where the server

11   is networked to other servers and clients.

12   In contrast, Lecourtier *et al.* U.S. Patent No. 6,373,850

13   ("Lecourtier"), assigned on its face to Bull S.A., discloses a

14   videoconferencing system that utilizes a standalone routing

15   switch. The Lecourtier system incorporates at least one group

16   switching center that transmits data directly, in a point-to-

17   point manner, to each data terminal that sends or receives the

18   video and data transmission. According to Lecourtier, this

19   method eliminates collisions between data terminals and

20   thereby allows large quantities of video and audio data to be

21   transmitted over long distances.

22   Howell U.S. Patent No. 5,767,897 ("Howell"), assigned on

23   its face to PictureTel Corporation, discloses a video

1   conferencing system that utilizes a director controller

2   situated at the site of the presentation (e.g., on a lecture

3   podium). The director controller includes a video display and

4   a control section. The video display allows the podium

5   speaker to view the transmitted video signals, whereas the

6   control section allows the podium speaker to selectively

7   control the distribution of generated audio and video

8   information signals among local and remote sites. The Howell

9   system incorporates networked controllers having a video

10   camera, microphone, and speaker that captures the podium

11   speaker's presentation and transmits the captured audio and

12   video to remote, networked locations also having a video

13   camera, microphone, speaker, and video display unit.

14   Yuan *et al.* U.S. Patent No. 5,821,986 ("Yuan"), assigned

15   on its face to PictureTel Corporation, discloses a scalable

16   video conferencing system for use between personal computers

17   connected via a network. The Yuan system encodes an image

18   sequence for transmission via the network. The type of

19   encoding disclosed in Yuan enables the image sequence to be

20   decoded at any one of at least two spatial resolutions and at

21   any one of the available frame rates. Such encoding allows

22   video transmission via a network environment in which there is

1   no guaranteed bandwidth or connection quality, such as via

2   standard twisted pair telephone lines.

3   Salesky *et al.* U.S. Patent No. 6,343,313 ("Salesky"),

4   assigned on its face to Pixion, Inc., discloses a computer

5   conferencing system with real-time, multipoint, multi-speed,

6   multi-stream scalability. The Salesky system is a server-

7   based system that allows conference participants to share all

8   or a portion of the video displayed on the participant's

9   computer screen. Furthermore, this system allows conference

10   participants to be located at sites that are remote from each

11   other, and requires all equipment, including the presenter's

12   equipment, to be connected to the same network.

13   Scherpbier U.S. Patent No. 6,263,365 ("Scherpbier"),

14   assigned on its face to Raindance Communications, Inc.,

15   discloses a browser controller that allows a pilot computer to

16   display selected web pages to both the pilot computer and one

17   or more passenger computers (e.g., to demonstrate and/or

18   discuss selected web page content during a conference call).

19   Initially, the user of the passenger computer is directed to

20   log on to a control web site, which downloads an active

21   control, such as an applet, to the passenger computer. When

22   the pilot computer chooses to display a particular web site to

23   the passenger computers, it sends the web site's URL to the

1 control site, which retrieves and sanitizes (i.e., disables

2 the hyperlinks) the web page. Thereafter, the control site

3 causes the passenger computers' active control to download the

4 sanitized web page. However, the web site viewed by the pilot

5 computer contains active hyperlinks, which the pilot computer

6 user may click to activate. The control site senses these

7 hyperlink selections and causes the pilot computer and

8 passenger computers to be connected to the chosen, hyperlink

9 web site.

10 Jiang U.S. Patent No. 6,167,432 ("Jiang"), assigned on

11 its face to WebEx Communications, Inc., discloses a method for

12 creating peer-to-peer connections over a network to facilitate

13 conferencing among network users. The Jiang system allows a

14 designated location (e.g., a web site) on an interconnected

15 network (e.g., the Internet) to be set up such that a

16 conference may be created, and conference participants may be

17 easily joined. The Jiang system maintains the Internet

18 Protocol ("IP") addresses of the conference participants at

19 the designated location. These addresses are transmitted to

20 new conference participants without requiring that the new

21 conference participants, or their equipment, have prior

22 knowledge of the IP addresses. Once the new participant is

23 joined in the conference, the data packets containing the

1   conference data do not pass through the designated location or

2   a central host, but rather are sent and received directly by

3   the application program of each participant.

4   Thus, in light of the prior art discussed herein, there

5   is a clear need for a standalone, multicast capable, video

6   transmission system that operates without the need for a

7   service provider and does not rely on a network server or the

8   Internet to control transmission of the video signals. Such a

9   system should provide near real time transmission of

10   compressed video. The video compression must be efficient

11   enough to transmit video in near real-time over standard

12   twisted pair telephone line bandwidths, yet it must not be too

13   lossy (i.e., the transmitted video images must not be

14   noticeably degraded). Further, the video transmission system

15   should be compatible with multiple platforms (e.g. Macintosh,

16   IBM, UNIX, etc.) and should not require any modifications,

17   including connection to a network, to the computer or other

18   video device used for the presentation. Also, the system

19   should not require a client computer at the participants'

20   locations. Finally, the system should have the ability to

21   control all video and data transmissions.

22

23

1 **SUMMARY OF THE INVENTION**

2   The present invention provides an improved video

3 transmission system that digitizes and compresses video

4 signals and transmits these signals with other digitized

5 signals to one or more remote locations. The present

6 invention is directed to multicasting video signals generated

7 at an initiating computer along with other digitized signals

8 generated either by or independent of the initiating computer

9 to multiple recipients. These signals may be sent via a

10 direct connection, LAN, WAN, Internet, or any other

11 communication medium.

12   One application of the present invention, and the

13 application that will be referred to herein to describe the

14 present invention in greater detail, is multicasting a

15 presentation performed on a local, initiating computer to

16 multiple remote recipients via remote display devices. The

17 transmitted video signals may include the signals displayed on

18 the initiating computer's screen (e.g., a PowerPoint®

19 presentation), video images received by a video device

20 connected to the initiating computer (e.g., a video camera),

21 or both. In this application, the same communication medium

22 that transmits the video signals also transmits other data

23 signals including IP address signals, session key signals,

1 keyboard signals, and optionally applet signals. These

2 signals are used by the video transmission system of the

3 present invention to authenticate and authorize presentation

4 participants. Presentations may be transmitted for a variety

5 of purposes, including educational purposes such as

6 distributing a lecture occurring at an educational

7 institution, to one or more remote locations.

8 Although the invention will be discussed herein with

9 respect to the aforementioned application, the present

10 invention may be used in any application in which video

11 signals and/or data signals are multicast to multiple

12 recipients. For example, the present invention may be used

13 for military applications. Drone airplanes currently transmit

14 telemetry data and streaming video received from a video

15 camera to a server on the ground. Thereafter, the server

16 multicasts the streaming video, but not the telemetry data.

17 By utilizing the present invention, both streaming video and

18 telemetry data may be multicast simultaneously to all

19 recipients.

20 It should also be noted that although the present

21 invention is discussed herein with respect to multicasting

22 video and other signals, the present invention may also be

23 used to broadcast these signals or transmit these signals in a

1 point-to-point manner. Broadcasting, as compared to

2 multicasting, simply requires the present invention to allow

3 all recipients, not just those that are authorized and

4 authenticated, to receive the broadcast signals. Point-to-

5 point transmission, as compared to multicasting, simply

6 requires that the video signals be sent to each recipient

7 individually, in lieu of sending the video signals to multiple

8 recipients simultaneously. Therefore, although the term

9 multicast is used herein for simplicity, it should be

10 understood that the present invention, as described herein,

11 may be used to broadcast signals, perform point-to-point

12 transmission of signals, and transmit signals using any other

13 non-multicast methods.

14 Minimally, the present invention may be implemented with

15 an initiating computer (i.e., a laptop or other computer

16 operated by the person performing the presentation), one video

17 server, a network (e.g., the Internet, LAN, WAN, etc.), and

18 one networked computer. In this minimal configuration, the

19 video out port (e.g., VGA port, SVGA port, etc.) of the

20 initiating computer, which may be any standard, unmodified

21 computer, is connected to the video in port of the video

22 server of the present invention via a standard video cable.

23 The video server is connected to the network using a standard

1 network cable (e.g., a CAT 5 cable) connected from the video

2 server network port to a port on the network, typically a port

3 of a network hub or router. The networked computer does not

4 require any hardware modifications as it is already connected

5 to the network.

6   In this minimal embodiment, the video server receives

7 video signals from the initiating computer and multicasts them

8 to the network via the network cable. When the networked

9 computer attempts to receive the multicast video signals from

10 the network, if the computer does not have the client software

11 of the present invention, the video server will automatically

12 download this software to the networked computer via the

13 network. After the software is installed, the video server

14 will attempt to authenticate and authorize this networked

15 computer to receive the video signals. If the attempt is

16 successful, the networked computer uses the downloaded client

17 software to receive and display the video signals multicast by

18 the video server via the network.

19   This minimal embodiment may be expanded to allow the

20 video transmission system user to display the presentation on

21 a video display device local to the initiating computer. In

22 this embodiment, a custom Y-style video cable is used in lieu

23 of the standard video cable to connect both the initiating

1   computer and a local display device to the video server. In

2   this application, the presentation video signals are provided

3   to the local video display device directly from the initiating

4   computer -- not the video server. However, the video server

5   does provide the same video signals to all remote video

6   display devices. This application requires a keyboard and/or

7   mouse to be connected to the video server to allow local

8   presentation participants to enter the data required for

9   authentication and authorization by the video server. The

10   users enter this data in response to prompts displayed on the

11   local video display device. The prompt data video signals are

12   generated by the video server of the present invention and

13   transmitted via the custom Y cable from the video server to

14   the local video display. That is, the custom Y cable

15   transmits either the presentation video signals from the

16   initiating computer to the local video display device or the

17   prompt data video signals from the video server to the local

18   video display device. The ability of the video server of the

19   present invention to directly accept a keyboard and a mouse

20   allows the presentation to be displayed locally without the

21   need for a computer, such as a networked computer, at the

22   local presentation site.

1   Similarly, a video client of the present invention may be

2   employed at remote presentation sites to work in conjunction

3   with the video server, thereby eliminating the need for a

4   computer at the remote site. Similar to the video server, a

5   video display device, keyboard, mouse, and system cable may be

6   directly connected to the video client. As discussed above

7   for the video server, direct connection of a keyboard and

8   mouse to the video client allows the remote presentation

9   participant to enter authentication and authorization

10   information in response to prompts displayed on the remote

11   video display device. These keyboard or mouse signals are

12   received at the video client and transmitted to the video

13   server, wherein an embedded web server, or other similar

14   device, makes all authentication and authorization decisions.

15   In the preferred embodiment of the present invention, the

16   video client's system cable is a CAT 5 cable, although any

17   type of cable may be used in any one of at least three

18   configurations. In a first configuration, the system cable

19   may connect the video client to a network, such as the

20   Internet or a LAN/WAN, and the network communicates the

21   multicast video signals from the video server to the video

22   client and its attached video display device. This first

23   configuration reduces wiring cost substantially by taking

644-033

1   advantage of existing network wiring. However, failure of a

2   network hub (i.e., a central connecting device in a network

3   that joins communications lines) may prevent the presentation

4   from reaching one or more remote participant locations.

5   Alternatively, in a second configuration, the system cable may

6   connect the video server directly to the video client. In

7   contrast to the first configuration, this second configuration

8   eliminates any dependence of the system on the network

9   equipment. That is, if the network fails, the multicast video

10   signals will still reach the remote video display device.

11   However, this second configuration increases the cost

12   associated with wiring the video server to one or more remote

13   participant locations. In a third, hybrid configuration, some

14   of the remote participant locations utilize direct system

15   cables and others utilize networked system cables. The

16   configuration of each remote participant location may be

17   chosen based upon the distance from the video server and the

18   criticality of failure. For example, a second or third

19   conference room may be wired to the video server via a direct

20   cable to ensure reliability in a location having a plurality

21   of participants. In contrast, a location having one

22   participant may be wired to the video server via a network

23   cable and, if a failure occurs, the individual may relocate to

644-033

1   a main conference room, wherein the video display device has a

2   direct connection to the video transmission system.

3   Remote participant locations may also be connected in a

4   hybrid configuration. That is, remote locations may include

5   networked computers operating client software automatically

6   downloaded to it from the video server, video clients attached

7   with a networked or direct system cable, or any combination of

8   the two.

9   Additionally, the present invention allows video signals

10   to be received from an initiating computer operating any one

11   of a variety of operating systems. Also, the remote display

12   devices and/or networked computers may operate with any one of

13   a variety of operating systems or protocols, including those

14   that differ from the initiating computer and/or other remote

15   networked computers. These operating systems and protocols

16   may include, but are not limited to, those manufactured by

17   Microsoft Corporation ("Microsoft") (Windows), Apple Computer,

18   Inc. ("Apple") (Macintosh), Sun Microsystems, Inc. ("Sun")

19   (Unix), Digital Equipment Corporation ("DEC"), Compaq Computer

20   Corporation ("Compaq") (Alpha), IBM (RS/6000), Hewlett-Packard

21   Company ("HP") (HP9000), SGI (formerly "Silicon Graphics,

22   Inc."), etc. Additionally, local devices may communicate with

23   remote computers via a variety of protocols including, but not

1   limited to, USB, American Standard Code for Information

2   Interchange ("ASCII"), Recommend Standard-232 ("RS-232"), etc.

3   The video server digitizes and compresses the video

4   signals received from the initiating computer prior to

5   transmission to the remote computers and/or remote video

6   display devices using the analog-to-digital conversion circuit

7   and compression algorithm disclosed in co-pending application

8   serial no. 10/233,299. The analog-to-digital conversion

9   circuit converts the received analog video signals into a

10   frame of digital pixel values using an analog-to-digital

11   ("A/D") converter. This circuit captures video output from

12   the initiating computer at a speed of at least 20

13   frames/second and converts the captured analog video signals

14   to a digital representation of pixels. Each pixel is

15   digitally represented with 5 bits for red, 5 bits for green,

16   and 5 bits for blue. The digital representation is then

17   stored in a raw frame buffer. The compression algorithm then

18   processes the digital data contained in the raw frame buffer.

19   The compression algorithm is actually a combination of four

20   sub-algorithms (i.e., the Noise Reduction and Difference Test

21   ("NRDT"), Smoothing, Caching, and Bit Splicing/Compression

22   sub-algorithms) as described in greater detail below.

1 After the video signals have been processed by the video

2 server and are transmitted to the remote participant location,

3 decompression occurs. That is, at the point that

4 decompression of the video signals is performed, the video

5 server has received the video signals from the initiating

6 computer, converted them from analog to digital, compressed

7 them (including reduction of noise), and transmitted them from

8 the video server to the remote participant's location. At the

9 remote participant location, either a video client or a

10 networked computer receives the video signals, decompresses

11 them, constructs the video image using the decompressed data

12 and data contained in its video cache, and transmits the

13 constructed video image to the remote participants' video

14 display device. The same decompression algorithm may be

15 modified to also receive transmitted data or control signals,

16 such as authentication and authorization signals produced by a

17 keyboard or mouse attached to the video server. Such signals

18 are required initially to allow the remote participant

19 location to be authorized to participate in the presentation.

20 Alternatively, these data or control signals may be processed

21 by hardware or software that operates independent of the

22 decompression algorithm.

1　　　　The remote equipment (i.e., video client, remote

2　networked computer, standalone computer, etc.) operates as a

3　decompression device by executing a decompression algorithm.

4　Along with any transmitted video or data signals, the video

5　server transmits messages to the decompression devices

6　regarding the portions of the video that yielded cache hits.

7　In response, the decompression device constructs the video

8　frame based upon the transmitted video signals and the blocks

9　of pixels contained in its local cache. Also, the

10　decompression device updates its local cache with the new

11　blocks of pixels received from the video server. In this

12　manner, the decompression device caches remain synchronized

13　with the compression device cache. Both the compression

14　device and the decompression device update their respective

15　cache by replacing older video data with newer video data.

16　　　　Furthermore, the video signals transmitted by the video

17　server have been compressed using a lossless compression

18　algorithm as discussed above. Therefore, the decompression

19　device must reverse this lossless compression. This is done

20　by identifying the changed portions of the video image, based

21　upon flags transmitted by the video server. From this flag

22　information, the decompression device is able to reconstruct

23　full frames of video.

1   In addition, the decompression device converts the video

2   frame to its original color scheme by reversing the CCT

3   conversion. Therefore, the decompression device, like the

4   video server, locally stores a copy of the same CCT used to

5   compress the video data. The CCT is then used to convert the

6   video data received from the video server to a standard RGB

7   format that may be displayed on the remote video display

8   device.

9   The decompression algorithm can be implemented in the

10   video transmission system of the present invention in a

11   variety of embodiments. For example, in one embodiment, it

12   can be implemented as a software application that is executed

13   by a remote networked or standalone computer. In this

14   embodiment, the video server may be configured to

15   automatically download software containing the decompression

16   algorithm to the remote computer during their initial

17   communication.

18   In an alternate embodiment, the decompression algorithm

19   can be implemented to execute within a web browser such as

20   Internet Explorer or Netscape® Navigator®. Such an embodiment

21   would eliminate the need for installation of application

22   specific software on the remote computer. Also, this

23   embodiment allows the video server to easily transmit the

- 25 -

1 presentation video signals to any computer with Internet

2 capabilities, regardless of the distance at which the computer

3 is located from the initiating computer. This feature reduces

4 the cabling cost associated with the video transmission system

5 of the present invention.

6   Finally, in yet another embodiment, the decompression

7 algorithm can be implemented in a device composed of a

8 microprocessor and memory, such as the video client of the

9 present invention. Such an embodiment completely eliminates

10 the need for a computer at the remote presentation location.

11   Since the present invention can be used to display

12 presentation video signals, or other types of video signals,

13 at locations that may be at a great distance from the

14 initiating computer, it is important to ensure that the video

15 signal transmission is secure. If the transmission is not

16 secure, hackers, competitors, or other unauthorized users

17 could potentially view the confidential presentation.

18 Therefore, the video transmission system of the present

19 invention is designed to easily integrate with digital

20 encryption techniques known in the art. In one embodiment of

21 the present invention, a 128-bit encryption technique is used

22 both to verify the identity of the remote participant and to

23 encrypt and decrypt the transmitted video and data signals.

1 In this embodiment, a 128-bit public key RSA encryption

2 technique is used to verify the remote participant, and a 128-

3 bit RC4 private key encryption is used to encrypt and decrypt

4 the transmitted signals. Of course, other encryption

5 techniques or security measures may be used.

6   Finally, since the video transmission system of the

7 present invention allows for platform independent

8 communications, the compression algorithm utilized by the

9 preferred embodiment of the present invention does not use

10 operating system specific hooks, nor does it employ platform

11 specific GDI calls.

12   In the preferred embodiment of the present invention, the

13 compression algorithm described herein and in co-pending

14 application serial no. 10/233,299 is used to transmit the

15 video signals. However, the video transmission system of the

16 present invention is not limited to such an embodiment.

17 Rather, this system may be employed with any compression

18 algorithm without departing from the spirit of the present

19 invention.

20   Therefore, it is an object of the invention to provide a

21 video multicast system for securely broadcasting video and/or

22 audio to remote participant equipment via a communications

1 network (e.g, the Internet, a local area network, a wide area

2 network, etc.)

3 It is also an object of the present invention to provide

4 a video multicast system that is expandable to accommodate any

5 number of participants.

6 It is another object of the invention to provide a video

7 multicast system capable of transmitting a video signal

8 without any video signal degradation.

9 It is another object of the invention to provide a video

10 multicast system which provides a secure and reliable

11 connection for all participants.

12 It is yet another object of the invention to provide a

13 video multicast system which is completely isolated from any

14 external networks.

15 It is still another an object of the invention to provide

16 a video multicast system in which the broadcast video signal

17 may be viewed utilizing software or a web browser.

18 It is yet another object of the invention to provide a

19 video multicast system for multicasting a video display to

20 conference rooms and/or individual computers located at a

21 corporation and over the Internet.

1   It is another object of the invention to provide a video

2   multicast system capable of multicasting a video signal as

3   well as telemetry data related to the video signal.

4   It is still another object of the invention to provide a

5   video multicast system for use in an educational environment.

6   It is still another object of the invention to provide a

7   video multicast system which allows video conferencing without

8   any setup or service provider.

9   It is another object of the invention to provide a vide

10 multicast system capable of displaying high quality video at

11 one or more locations remote from a presenter.

12 Other objects, features, and characteristics of the

13 present invention, as well as the methods of operation and

14 functions of the related elements of the structure, and the

15 combination of parts and economies of manufacture, will become

16 more apparent upon consideration of the following detailed

17 description with reference to the accompanying drawings, all

18 of which form a part of this specification.

19

20 **BRIEF DESCRIPTION OF THE DRAWINGS**

21 A further understanding of the present invention can be

22 obtained by reference to the preferred embodiment as well as

23 some alternate embodiments set forth in the illustrations of

1 the accompanying drawings. Although the illustrated

2 embodiments are merely exemplary of systems for carrying out

3 the present invention, the organization, expanded

4 configurations and method of operation of the invention, in

5 general, together with further objectives and advantages

6 thereof, may be more easily understood by reference to the

7 drawings and the following description. The drawings are not

8 intended to limit the scope of the invention, which is set

9 forth with particularity in the claims as appended or as

10 subsequently amended, but merely to clarify and exemplify the

11 invention.

12 For a more complete understanding of the present

13 invention, reference is now made to the following drawings in

14 which:

15 FIG. 1 depicts a block diagram of the hardware

16 configuration of the preferred embodiment of the multicast

17 system according to the present invention in which the video

18 transmission system is utilized to multicast presentation

19 video signals from an initiating computer to multiple remote

20 participant locations.

21 FIG. 2 depicts a block diagram of the preferred

22 embodiment of the video server shown in FIG. 1 including the

1 interface to the initiating computer and the communications

2 device for interfacing to the remote participant locations.

3 FIG. 3A depicts a block diagram of a remote computer

4 which, in one embodiment of the invention, performs the

5 decompression according to the invention.

6 FIG. 3B depicts a block diagram of an independent

7 hardware device referred to herein as the video client, which,

8 in another embodiment of the invention, performs the

9 decompression according to the invention.

10 FIG. 4 depicts a flowchart of the compression algorithm

11 utilized by the preferred embodiment of the present invention.

12 FIG. 5A depicts a detailed flowchart of the NRDT and

13 smoothing sub-algorithms of the compression algorithm shown in

14 FIG. 4.

15 FIG. 5B depicts a detailed flowchart of the caching and

16 bit splicing/compression sub-algorithms of the compression

17 algorithm shown in FIG. 4.

18 FIG. 6 depicts a detailed flowchart of the nearest color

19 match function of FIG. 5A and its integration with the CCT of

20 the compression algorithm.

21 FIG. 7A depicts a detailed flowchart of the RGB NRDT sub-

22 algorithm of FIG. 5A.

1   FIG. 7B depicts an example application of the NRDT sub-

2 algorithm to a sample block of pixels as performed by the

3 compression algorithm of the present invention.

4   FIG. 8 depicts a detailed flowchart of the operation of

5 the decompression algorithm according to the preferred

6 embodiment of the present invention.

7

8 **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

9   As required, a detailed illustrative embodiment of the

10 present invention is disclosed herein. However, systems and

11 operating structures in accordance with the present invention

12 may be embodied in a wide variety of forms and modes, some of

13 which may be quite different from those in the disclosed

14 embodiment. Consequently, the specific structural and

15 functional details disclosed herein are merely representative,

16 yet in that regard, they are deemed to afford the best

17 embodiment for purposes of disclosure and to provide a basis

18 for the claims herein which define the scope of the present

19 invention. The following presents a detailed description of a

20 preferred embodiment (as well as some alternative embodiments)

21 of the present invention.

22   Referring first to FIG. 1, depicted is a block diagram of

23 the hardware configuration of the video multicast system

1   according to the preferred embodiment of the present

2   invention. The term "local" will be used herein to describe

3   the area proximal to video server 102, and the term "remote"

4   will be used herein to describe participant locations that are

5   connected to video server 102 via a connection other than

6   communication path 112 or its equivalent.

7   As illustrated in FIG. 1, the video signals generated by

8   initiating computer 104 are transmitted from standard video

9   out port 152 via communication path 112 to video in port 154

10 of video server 102 and video in port 156 of video display

11 device 110. Video out port 152 and video in port 156 are

12 standard video ports that come pre-packaged with initiating

13 computer 104 and video display device 156, respectively. In

14 the preferred embodiment, communication path 112 includes two

15 VGA connectors for connection to these ports. However,

16 communication path 112 may be alternatively configured to

17 include ports other than VGA including, but not limited to,

18 SuperVGA ("SVGA"), S-video, composite video, etc.

19   Preferably, video server 112 also contains mouse port 158

20 and keyboard port 160, which connect mouse 106 and keyboard

21 108, respectively, to video server 112. Also, in the

22 preferred embodiment, mouse 106 and keyboard 108 are connected

23 via separate communication paths 162 and 164, respectively,

1   and many types of such paths (e.g., cables, wireless, etc.)

2   are well known in the art. However, any method of connecting

3   mouse 106 and keyboard 108 to video server 102 may be used

4   with the present invention. For example, one alternative

5   configuration connects mouse 106 and keyboard 108 to video

6   server 102 via a shared USB connection. In this

7   configuration, mouse port 158 and keyboard port 160 are

8   actually one combined USB port. In another configuration,

9   mouse 106 and keyboard 108 connect to video server 102 via a

10 wireless connection. Alternatively, depending on the desired

11 application of the present invention, either or both of mouse

12 106 and keyboard 108 may not be required for system operation.

13 Mouse 106 and/or keyboard 108 are utilized to allow the local

14 presentation participants to provide input to video server

15 102. This feature allows the presentation to be interactive.

16 For example, participants may provide feedback regarding the

17 speed of the presentation (i.e., too fast or too slow),

18 readiness, questions, etc.

19   In an alternative embodiment, the system user may not

20 need or intend to display the presentation locally. In such

21 an embodiment, communication path 112 may be configured to

22 connect initiating computer 104 to video server 102 only.

1 Here, video display device **110**, mouse **106**, and keyboard **108**

2 are not required.

3 Video server **102** includes communication devices **118** and

4 **120**, and video client **124** and remote computer **138** include

5 communication devices **122** and **124**, respectively. Each of

6 these communication devices is capable of bi-directional,

7 digital communication via its attached communication path **114**

8 or **116**. Communication devices **118, 120, 122,** and **124** may be

9 modems, network interface cards, wireless network cards, RS-

10 232 transceivers, RS-485 transceivers, or any similar devices

11 capable of providing bi-directional digital communication

12 signals. Similarly, communication paths **114** and **116** may be

13 standard telephone lines, the Internet, wireless connections,

14 a LAN, a WAN, cables, or any other medium capable of

15 transmitting bi-directional digital communication signals.

16 Communication devices **118** and **120** enable video server **102** to

17 communicate with video client **124** and remote computer **138**,

18 respectively, via any standard agreed upon protocol. Examples

19 of these protocols include, but are not limited to,

20 Transmission Control Protocol/ Internet Protocol ("TCP/IP"),

21 Ethernet, User Datagram Protocol ("UDP"), etc.

22 Communication device **122** of video client **124** receives the

23 digitized and compressed video signals transmitted via

1 communication path **114** from video server **102**. Thereafter, the

2 transmitted video signals are decompressed via a decompression

3 algorithm that corresponds with the compression algorithm

4 utilized by video server **102**. After decompression, the video

5 signals are translated, if necessary, to match the video

6 format of attached video display device **126**. Finally, the

7 video signals are transmitted to video display device **126** via

8 video port **132** allowing the remote participants to view the

9 presentation being demonstrated on initiating computer **104**.

10 In addition to receiving video signals from video server

11 **102**, video client **124** also receives signals from keyboard **128**

12 and mouse **130** via keyboard port **134** and mouse port **136**,

13 respectively. A system user located at the remote video

14 client site utilizes keyboard **128** and/or mouse **130** to enter

15 input. This input may be the same as that described above for

16 video server **102** or, alternatively, the input may correspond

17 to authentication and authorization data. In the preferred

18 embodiment, the system user or remote participant enters this

19 data in response to prompts displayed on video display device

20 **126**. The prompt data video signals are generated by video

21 server **102** and transmitted via communication path **114** from

22 video server **102** to video display device **126**. That is,

23 communication path **114** transmits either the presentation video

1   signals generated by initiating computer 104 or prompt data

2   video signals generated by video server 102 to video display

3   device 126. Additionally, communication path 114 transmits

4   system user input generated by mouse 128 and/or keyboard 130

5   from video client 124 to video server 102. Initially, video

6   server 102 utilizes this information to determine whether

7   video client 124 may receive the presentation video signals.

8   Thereafter, this user input allows the presentation to be

9   interactive as discussed above with respect to mouse 106 and

10   keyboard 108. The ability of video client 124 to directly

11   accept signals from keyboard 128 and mouse 130 allows the

12   presentation to be displayed remotely without the need for a

13   computer at the remote presentation site.

14   However, in situations where it is desired to display the

15   presentation at a location already equipped with a computer,

16   such as remote computer 138, the system does not require a

17   second video client 124. Rather, remote computer 138 receives

18   video signals directly from video server 102 via communication

19   path 116. Initially, when a remote participant using remote

20   computer 138 attempts to receive the multicast video signals,

21   video server 102 automatically downloads client software to

22   remote computer 138 via communication path 116. After the

23   software is installed, video server 102 will attempt to

1 authenticate and authorize remote computer **138**. If the

2 attempt is successful, the remote computer **138** receives the

3 multicast video signals from video server **102** via

4 communication path **116**.

5   According to the invention, any combination of video

6 clients **124** may be combined with any number of remote

7 computers **138**. Additionally, each one of these devices may be

8 connected to video server **102** via its own type of

9 communication path **114** or **116**. In other words, some video

10 clients **124** or remote computers **138** may be connected to video

11 server **102** via a corporate LAN or WAN, and these networked

12 devices may operate with non-networked video clients **124** and

13 remote computers **138** that are connected via direct cabling to

14 form a single video transmission system.

15   Turning to FIG. 2, depicted is a block diagram of the

16 circuitry of video server **102** shown in FIG. 1 according to the

17 preferred embodiment of the invention. Preferably, video

18 server **102** comprises A/D converter **201**, pixel pusher **221**,

19 frame buffers **227**, JBIG **229**, flash memory **239**, RAM **241**,

20 microprocessor **223**, video-in port **154**, Dual universal

21 asynchronous receiver transmitter ("DUART") **233**, mouse port

22 **158**, keyboard port **160**, clock **213**, and communications devices

23 **118** and **120**.

1 A/D converter **201** receives video signals from initiating

2 computer **104** via communication path **112** through video in port

3 **154**. The first step in preparing the video for transmission

4 is conversion of the video signals from analog signals to

5 digital signals, which is performed by A/D converter **201**. A/D

6 converter **201** receives analog red signals, analog green

7 signals, analog blue signals, horizontal synchronization

8 signals, and vertical synchronization signals via conductors

9 **203**, **205**, **207**, **209**, and **211**, respectively. Clock **213** drives

10 the timing of A/D converter **201** and microprocessor **223** using

11 means commonly employed in the art. The outputs of A/D

12 converter **201** are shown as R-out **215**, G-out **217**, and B-out

13 **219**, which represent the red component, green component, and

14 blue component of the digitized video signal, respectively.

15 A/D converter **201** outputs these digitized video signal

16 components in the form of pixels, which are transmitted to and

17 stored in pixel pusher **221**. Pixel pusher **221**, flash memory

18 **239**, and Random Access Memory ("RAM") **241** communicate with

19 microprocessor **223** via communication bus **225**. Pixel pusher

20 **221** also communicates with frame buffers **227** (e.g., raw frame

21 buffer, compare frame buffer, etc.) and compression device **229**

22 via communication buses **243** and **245**, respectively. The

1 compression algorithm is executed by microprocessor **223**.

2 Generally, the compression algorithm operates as follows:

3

4 **Noise Reduction and Difference Test:**

5   As discussed above, digitization of the analog video

6 signals is necessary to allow these signals to be transmitted

7 via a digital communication medium (e.g., a network, LAN, WAN,

8 Internet, etc.). However, a detrimental side effect of the

9 digitization process is the introduction of quantization

10 errors and noise into the video signals. Therefore, the Noise

11 Reduction and Difference Test sub-algorithm ("NRDT sub-

12 algorithm") is designed to reduce the noise introduced during

13 the digitization of the video signals. In addition, the NRDT

14 sub-algorithm simultaneously determines the differences

15 between the recently captured frame of video (i.e., the

16 "current frame") and the previously captured frame of video

17 (i.e., the "compare frame").

18   First, the NRDT sub-algorithm divides the current frame,

19 which is contained in the raw frame buffer, into 64 x 32

20 blocks of pixels. Alternatively, other sizes of blocks may be

21 used (e.g., 8x8 pixels, 16x16 pixels, 32x32 pixels, etc.)

22 based upon criteria such as the size of the entire video

1 frame, the bandwidth of the communication medium, desired

2 compression yield, etc.

3   After the current frame is divided into blocks, a two-

4 level threshold model is applied to the block of pixels to

5 determine whether it has changed with respect to the compare

6 frame. These two thresholds are the pixel threshold and the

7 block threshold.

8   First, a given pixel is examined and the value of each of

9 the three colors (i.e., red, green, and blue) of the pixel is

10 calculated with the value of its corresponding pixel in the

11 compare frame. From this calculation, a distance value is

12 computed. If the distance value is greater than the pixel

13 threshold (i.e., the first threshold of the two-level

14 threshold), this distance value is added to a distance sum.

15 This process is performed for each pixel in the block.

16   Next, after the distance value of all of the pixels in

17 the block have been calculated and processed in the

18 aforementioned manner, the resulting value of the distance sum

19 is compared to the block threshold (i.e., the second threshold

20 of the two-level threshold). If the distance sum exceeds the

21 block threshold, then this block of pixels is considered

22 changed in comparison to the corresponding block of pixels in

23 the compare frame. If a change is determined, the compare

1 frame, which is stored in the compare frame buffer, will be

2 updated with the new block of pixels. Furthermore, the new

3 block of pixels will be further processed and transmitted in a

4 compressed format to the remote presentation equipment.

5 In contrast, if the distance sum is not greater than the

6 block threshold, the block of pixels is determined to be

7 unchanged. Consequently, the compare frame buffer is not

8 updated, and this block of pixels is not transmitted to the

9 remote presentation equipment. Eliminating the transmission

10 of unchanged blocks of pixels reduces the overall quantity of

11 data to be transmitted, thereby increasing transmission time

12 and decreasing the required bandwidth.

13 The NRDT sub-algorithm is ideal for locating both a large

14 change in a small quantity of pixels and a small change in a

15 large quantity of pixels. Consequently, the NRDT sub-

16 algorithm is more efficient and more accurate than known

17 percentage threshold algorithms that simply count the number

18 of changed pixels in a block of pixels. With such an

19 algorithm, if a few pixels within the block of pixels have

20 changed drastically (e.g., from black to white), the algorithm

21 would consider the block of pixels to be unchanged since the

22 total number of changed pixels would not exceed the percentage

1 threshold value. This result will often lead to display

2 errors in the transmission of computer video.

3   Consider, for example, a user that is editing a document.

4 If the user were to change a single letter, such as changing

5 an "E" to an "F," only a few pixels of the video image would

6 change. However, based upon this change, the resulting

7 document is dramatically different than the original document.

8 A percentage threshold algorithm would not register this

9 change and, therefore, would lead to a display error. A

10 percentage threshold algorithm, by only looking at the number

11 of pixels within a block that have changed, generally fails to

12 recognize a video image change in which a few pixels have

13 changed substantially. However, the NRDT sub-algorithm used

14 by the present invention, by virtue of its two-level

15 threshold, will recognize that such a block of pixels has

16 significantly changed between successive frames of video.

17

18 **Smoothing:**

19   When the NRDT sub-algorithm determines that a block of

20 pixels has changed, the digital data that represents this

21 block is further processed by a smoothing sub-algorithm. This

22 sub-algorithm reduces the roughness of the video image that is

1   caused by the noise introduced during the analog-to-digital

2   conversion.

3   First, each digital pixel representation is converted to

4   a representation that uses a lower quantity of bits for each

5   pixel. It is known in the art to compress color video by

6   using a fewer number of bits to represent each color of each

7   pixel. For example, a common video standard uses 8 bits to

8   represent each of the red, green, and blue components of a

9   video signal. This representation is commonly referred to as

10 an "8 bit RGB representation". If only the four most

11 significant bits of the red, green, and blue components of the

12 pixel are used to represent its color in lieu of all eight

13 bits, the total amount of digital data used to represent the

14 block of pixels, and frame of video, is reduced by fifty

15 percent.

16   In contradistinction, the smoothing sub-algorithm

17 incorporates a more intelligent method of reducing the size of

18 an RGB representation. This method uses a Color Code Table

19 ("CCT") to map specific RGB representations to more compact

20 RGB representations. Both the compression and decompression

21 algorithms of the present invention use the same CCT.

22 However, different color code tables may be chosen depending

1 on the available bandwidth, the capabilities of the local

2 display device, etc.

3   For each block of pixels, a histogram of pixel values is

4 created and sorted by frequency such that the smoothing sub-

5 algorithm may determine how often each pixel value occurs.

6 Pixel values that occur less frequently are compared to pixel

7 values that occur more frequently. To determine how similar

8 pixel values are, a distance value is calculated based upon

9 the color values of the red, green, and blue ("RGB")

10 components of each pixel. During the histogram analysis, a

11 map of RGB values to color codes (i.e., a CCT) is created. If

12 a less frequently occurring pixel value needs to be adjusted

13 to a similar, more frequently occurring pixel value (i.e.,

14 color), the CCT is used to map the less frequently occurring

15 pixel value to the color code of the more frequently occurring

16 pixel value. Thus, the noise is efficiently removed from each

17 block and the number of bits used to represent each pixel is

18 reduced.

19   For illustrative purposes, suppose that an 8x8 pixel

20 block is being processed. Further suppose that of the 64

21 pixels in the current block, 59 are blue, 4 are red, and 1 is

22 light blue. Further assume that a low frequency threshold of

23 5 and a high frequency threshold of 25 are used. In other

1   words, if a pixel value occurs less than 5 times within a

2   block, it is considered to have a low frequency. Similarly,

3   if a pixel value occurs more than 25 times within a block, it

4   is considered to have a high frequency. In the preferred

5   embodiment of the present invention, the smoothing sub-

6   algorithm ignores pixel values occurring between these two

7   thresholds. Therefore, in the present example, the smoothing

8   sub-algorithm determines that the red and light blue pixels

9   have a low frequency, and the blue pixels have a high

10 frequency.

11   In the next step, the values of the 4 red pixels and the

12 1 light blue pixel are compared with the value of the blue

13 pixels. In this step, a pre-determined distance threshold is

14 used. If the distance between the less frequent pixel value

15 and the more frequent pixel value is within this distance

16 threshold, then the less frequent pixel value is converted to

17 the more frequent pixel value. Therefore, in our present

18 example, it is likely that the light blue pixel is close

19 enough in value to the blue pixel that its distance is less

20 than the distance threshold. Consequently, the light blue

21 pixel is mapped to the blue pixel. In contrast, it is likely

22 that the distance between the red and blue pixels exceeds the

23 distance threshold and, therefore, the red pixel is not mapped

1 to the blue pixel. With the smoothing sub-algorithm of the

2 present invention, although the red pixels occur rarely, the

3 distance between the red pixel value and the blue pixel value

4 is large enough that the red pixels are not converted to blue

5 pixels. In this manner, the smoothing sub-algorithm of the

6 present invention increases the redundancy in compared images

7 by eliminating changes caused by superfluous noise introduced

8 during the analog-to-digital conversion while retaining real

9 changes in the video image.

10

11 **Caching:**

12   After the smoothing sub-algorithm has been applied to the

13 digital video image data, an optional caching sub-algorithm

14 may be applied to further minimize the bandwidth required for

15 transmitting the video images. The caching sub-algorithm

16 relies on a cache of previously transmitted blocks of pixels.

17 Similar to the NRDT sub-algorithm, the caching sub-algorithm

18 is performed on a block of pixels within the video frame.

19 Again, any block size may be used (e.g., 8x8, 16x16, 32x32 or

20 64x32).

21   First, the caching sub-algorithm performs a cache check,

22 which compares the current block of pixels with blocks of

23 pixels stored in the cache. The cache may store an

1 arbitrarily large number of previous frames. Cache hits are

2 likely to occur a higher percentage of the time if a large

3 cache is incorporated. However, memory and hardware

4 requirements increase when the size of the cache is increased.

5 Furthermore, the number of comparisons, and thus the

6 processing power requirements, also increases when the size of

7 the cache increases.

8 A "cache hit" occurs when a matching block of pixels is

9 located within the cache. A "cache miss" occurs if a matching

10 block of pixels is not found in the cache. When a cache hit

11 occurs, the new block of pixels does not have to be

12 retransmitted. Instead, a message and a cache entry

13 identification ("ID") are sent to the remote participant

14 equipment. Generally, this message and cache entry ID will

15 consume less bandwidth than that required to transmit an

16 entire block of pixels.

17 If a "cache miss" occurs, the new block of pixels is

18 compressed and transmitted to the remote participant

19 locations. Also, both the remote and local devices update

20 their respective cache by storing the new block of pixels in

21 the cache. Since the cache is of limited size, older data is

22 overwritten. One skilled in the art is aware that various

23 algorithms can be used to decide which older data should be

1 overwritten. For example, a simple algorithm can be employed

2 to overwrite the oldest block of pixels within the cache,

3 wherein the oldest block is defined as the least recently

4 transmitted block.

5   In order to search for a cache hit, the new block of

6 pixels must be compared with all corresponding blocks of

7 pixels located within the cache. There are several ways in

8 which this may be performed. In one embodiment, a cyclic

9 redundancy check ("CRC") is computed for the new block of

10 pixels and all corresponding blocks of pixels. The CRC is

11 similar to a hash code for the block. A hash code is a

12 smaller, yet unique, representation of a larger data source.

13 Thus, if the CRCs are unique, the cache check process can

14 compare CRCs for a match instead of comparing the whole block

15 of pixels. If the CRC of the current block of pixels matches

16 the CRC of any of the blocks of pixels in the cache, a "cache

17 hit" has been found. Because the CRC is a smaller

18 representation of the block, less processing power is needed

19 to compare CRCs. Furthermore, it is possible to construct a

20 cache in which only the CRCs of blocks of pixels are stored at

21 the remote participant locations. Thus, using a CRC

22 comparison in lieu of a block of pixels comparison saves

23 memory and processor time, which results in a lower cost.

1 **Bit Splicing/Compression:**

2   Once the NRDT, smoothing, and optional caching sub-

3 algorithms are performed, each block of pixels that must be

4 transmitted is ready to be compressed. In the preferred

5 embodiment of the present invention, each block is compressed

6 using the Joint Bi-level Image Group ("JBIG") lossless

7 compression algorithm.

8   The JBIG compression algorithm was designed for black and

9 white images, such as those transmitted by facsimile machines.

10 However, the compression algorithm utilized by the present

11 invention compresses and transmits color video images.

12 Therefore, when utilizing the JBIG compression algorithm, the

13 color video image must be bit-sliced, and the resulting bit-

14 planes must be compressed separately.

15   A bit plane of a color video image is created by grabbing

16 a single bit from each pixel color value in the color video

17 image. For example, if 8 bits are used to represent the color

18 of the pixel, then the color video image is divided into 8 bit

19 planes. The compression algorithm works in conjunction with

20 the CCT discussed above to transmit the bit plane containing

21 the most significant bits first, the bit plane containing the

22 second most significant bits second, etc. The CCT is designed

23 such that the most significant bits of each pixel color are

1   stored first and the lesser significant bits are stored last.

2   Consequently, the bit planes transmitted first will always

3   contain the most significant data, and the bit planes

4   transmitted last will always contain the least significant

5   data. Thus, the remote video display devices will receive

6   video from the initiating computer progressively, receiving

7   and displaying the most significant bits of the image before

8   receiving the remaining bits. Such a method is less sensitive

9   to changes in bandwidth and will allow a user to see the frame

10 of video as it is transmitted, rather than waiting for all

11 details of the frame to be sent.

12   After compression of the video signals is complete, the

13 resulting video signals are transmitted via communication bus

14 **225** to communication devices **118** and **120**. Then, the digitized

15 and compressed video signals are transmitted via communication

16 devices **118** and **120** to the attached communication medium,

17 which transmits the video signals to the intended video

18 client(s) **124** or remote computer(s) **138** (FIG. 1).

19   Dual universal asynchronous receiver transmitter

20 ("DUART") **233** also communicates with microprocessor **223** via

21 communication bus **225**. DUART **233** also interfaces with

22 keyboard port **160** and mouse port **158**. Thus, DUART **233**

23 receives keyboard and mouse signals generated at keyboard **108**

1   and mouse **106** (FIG. 1) via keyboard port **160** and mouse port

2   **158,** whereupon they are processed by DUART **233** and transmitted

3   via communication bus **225** to microprocessor **223.**

4   Referring next to FIG. 3A, illustrated is a block diagram

5   of the internal circuitry of remote computer **138** shown in FIG.

6   1. As shown, remote computer **138** contains microprocessor **303,**

7   which executes the decompression algorithm (discussed in

8   further detail below with respect to FIG. 8). Remote computer

9   **138** additionally contains video card **311,** keyboard port **315,**

10   mouse port **319,** memory **307,** and communications device **124.**

11   Microprocessor **303** receives data (i.e., video signal data,

12   control data, etc.) from communication device **124** via

13   communication bus **305.** Communication device **124** may utilize

14   shared memory, a memory bus, and/or drivers to perform such

15   data transmission.

16   Communication device **124** receives data from video server

17   **102** via communications path **116** (FIG. 1). When a potential

18   presentation participant wishes to connect to a multicast

19   presentation via remote computer **138,** microprocessor **303**

20   executes the decompression algorithm, which is stored in

21   computer readable medium **307,** via communication bus **309.** The

22   decompression algorithm decompresses the data received from

23   communication device **124** and converts the data into a format

1 that is compatible with video card **311**. The properly formatted

2 data is then transmitted to video card **311** via communication

3 bus **313**, whereupon video signals are generated for display and

4 are transmitted via communication path **140** to video display

5 device **146**.

6   In addition, microprocessor **303** receives keyboard and

7 mouse signals. Keyboard signals generated at keyboard **148** are

8 transmitted via communication path **142** through keyboard port

9 **315** to communication bus **317** to microprocessor **303**. In a

10 similar manner, mouse signals generated at mouse **150** are

11 transmitted via communication path **144** through mouse port **319**

12 to communication bus **321** to microprocessor **303**.

13   Although the video transmission system of the present

14 invention allows an existing, networked or non-networked

15 personal computer to be used to display a multicast

16 presentation, a computer is not required. FIG. 3B depicts a

17 block diagram of the circuitry of the preferred embodiment of

18 video client **124** of the present invention. That is, video

19 client **124** has the ability to display the presentation on

20 video display device **126** without the need for a personal

21 computer. Video display unit **126**, keyboard **128**, and mouse **130**

22 receives signals from video port **325**, keyboard port **327**, and

1 mouse port 329 of video client 124 via communication paths

2 132, 134, and 136, respectively.

3 Communication device 122 receives data from video server

4 102 via communications path 114. When a potential

5 presentation participant wishes to connect to a multicast

6 presentation via video client 124, microprocessor 333 executes

7 the decompression algorithm, which is stored in memory 331,

8 via communication bus 338. The decompression algorithm

9 decompresses the data received from communication device 122

10 and converts the data into a format that is compatible with

11 video display unit 126. The properly formatted data is then

12 transmitted to video port 325 via communication bus 337,

13 whereupon video signals are generated for display and are

14 transmitted via communication path 132 to video display device

15 126.

16 In addition, microprocessor 333 receives keyboard and

17 mouse signals generated at keyboard 128 that are transmitted

18 via communication path 134, keyboard port 327, and

19 communication bus 337 to microprocessor 333. In a similar

20 manner, mouse signals generated at mouse 130 are transmitted

21 via communication path 136, mouse port 329, and communication

22 bus 337 to microprocessor 333.

1   Referring now to FIG. 4, depicted is a flowchart

2   illustrating the operation of the compression algorithm

3   utilized in the preferred embodiment of the present invention.

4   The compression algorithm is executed internal to video server

5   102 by microprocessor 223 (FIG. 2). Prior to processing the

6   video signal data, the video image must be captured (step

7   401). The data is captured from initiating computer 104 by

8   A/D converter 201 via video in port 154 (FIG. 2) at a speed of

9   at least 20 frames/second. A/D converter 201 and pixel pusher

10   221 then convert the analog video signals to a digital

11   representation of pixels, with each pixel represented as 5

12   bits for red, 5 bits for blue, and 5 bits for green.

13   Thereafter, this unprocessed pixel data is stored in a raw

14   frame buffer (step 402), which is one of the frame buffers 227

15   (FIG. 2). At this point, the compression algorithm is

16   performed to process the captured video data contained in the

17   raw frame buffer and prepare it for multicasting to remote

18   participant locations.

19   The first step of the compression algorithm is the NRDT

20   (step 403). The NRDT sub-algorithm is also executed internal

21   to video server 102 by microprocessor 223 (FIG. 2). The NRDT

22   sub-algorithm determines which blocks of pixels, if any, have

23   changed between the current frame and the compare frame.

1   In the preferred embodiment, the video frame is first

2   divided into 64x32 pixel blocks. Subsequently, the NRDT sub-

3   algorithm is applied to each block of pixels independently.

4   Alternative embodiments of the present invention may utilize

5   smaller or larger blocks depending on criteria such as desired

6   video resolution, available bandwidth, etc.

7   Next, the NRDT sub-algorithm employs a two-threshold

8   model to determine whether differences exist between a block

9   of pixels in the current frame and the corresponding block of

10   pixels in the compare frame. These two thresholds are the

11   pixel threshold and the block threshold.

12   First, each pixel of the pixel block is examined to

13   determine if that pixel has changed relative to the

14   corresponding pixel of the corresponding block in the compare

15   frame. The distance value of each of the three colors (i.e.,

16   red, green, and blue) of each pixel in relation to the

17   corresponding compare pixel is calculated, as described in

18   greater detail below with respect to FIG. 7. If the distance

19   value is larger than the pixel threshold (i.e., the first

20   threshold of the two-threshold model), this distance value is

21   added to a distance sum value.

22   Then, after all pixels within the pixel block have been

23   examined, if the resulting distance sum value is greater than

1   the block threshold (i.e., the second threshold of the two-

2   threshold model), the block is determined to have changed.

3   Every block of pixels in the video frame undergoes the same

4   process. Therefore, after this process has been applied to an

5   entire video frame, the process will have identified all pixel

6   blocks that the process has determined have changed since the

7   previous video frame. At this point, the compare frame is

8   updated with the changed pixel blocks. However, the pixel

9   blocks of the compare frame that correspond to unchanged pixel

10 blocks of the current frame will remain unchanged. In this

11 manner, the two-threshold model used by the NRDT sub-algorithm

12 eliminates pixel value changes that are introduced by noise

13 created during the analog to digital conversion and also

14 captures the real changes in the video frame.

15   After the video data is processed by the NRDT sub-

16 algorithm, it is next processed by the smoothing sub-algorithm

17 (step 419). The smoothing sub-algorithm is designed to create

18 a smooth, high-quality video image by reducing the roughness

19 of the video image caused by noise introduced during the

20 analog to digital conversion.

21   The smoothing sub-algorithm first converts the pixel

22 representation that resulted from the NRDT sub-algorithm into

23 a pixel representation that uses a lesser quantity of bits to

1   represent each pixel. This is performed using a CCT that is

2   specially organized to minimize the size of the pixel

3   representation. The smoothing sub-algorithm uses the CCT to

4   choose color codes with the least number of 1-bits for the

5   most commonly used colors. For example, white and black are

6   assumed to be very common colors. Thus, white is always

7   assigned 0 and black is always assigned 1. That is, white

8   will be represented by a bit value of 0 on all planes. Black,

9   the next most common color, will show up as a bit value of 0

10 on all but one plane. This reduces the quantity of data to be

11 compressed by the compression algorithm. Then, for each pixel

12 in the block, a color code is assigned. Simultaneously, a

13 histogram of color codes is created to store the number of

14 occurrences of each of the unique colors in the block of

15 pixels. This histogram of color codes is then sorted to

16 produce a list of color codes from the least number of

17 occurrences to the dominant number of occurrences.

18   Once the sorted list of color codes is created, the next

19 step is to merge colors. Working from the beginning of the

20 sorted list, the smoothing sub-algorithm compares the least

21 frequently occurring colors to the more frequently occurring

22 colors. If the less frequently occurring color is very

23 similar to a more frequently occurring color, then the pixels

1   having the less frequently occurring color will be changed to

2   the more frequently occurring color. Determination of whether

3   two colors are similar is performed by calculating the

4   distance between the three-dimensional points of the RGB

5   space. The formula is:

6   $$D = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}$$

7   where D is the distance, R_1 is the red value of the low

8   frequency pixel, R_2 is the red value of the high frequency

9   pixel, G_1 is the green value of the low frequency pixel, G_2 is

10 the green value of the high frequency pixel, B_1 is the blue

11 value of the low frequency pixel, and B_2 is the blue value of

12 the high frequency pixel.

13   If the distance is within a distance threshold, the two

14 colors are determined to be similar. In the preferred

15 embodiment of the present invention, system performance is

16 increased by squaring the distance threshold and comparing

17 this value with the sum of the squares of the RGB differences.

18 This step eliminates taking the square root of the sum, which

19 requires a greater amount of processing time.

20   Each block of pixels is filtered for noise and translated

21 from a RGB representation to a color code representation. The

22 noise that is introduced by A/D converter **201** (FIG. 2) during

23 conversion of the analog signals to digital signals distorts

1 the values of some pixels. Thus, the smoothing sub-algorithm

2 is designed to recognize distorted pixels and adjust them to

3 the correct value. The smoothing sub-algorithm minimizes

4 noise by reducing the number of different colors present in

5 each video image block. Such smoothing creates an image with

6 greater redundancy, thus yielding higher compression ratios.

7 After smoothing, caching is performed (step **421**).

8 Caching is a sub-algorithm of the overall compression

9 algorithm executed by microprocessor **223** of video server **102**

10 (FIG. 2). Caching requires video server **102** (FIG. 2) to

11 retain a cache of recently transmitted images. Such a cache

12 can be implemented and stored in RAM **241** (FIG. 2). The

13 caching sub-algorithm compares the most recent block of pixels

14 with the corresponding block of pixels in the video images

15 stored in the cache (step **405**). If the most recently

16 transmitted block of pixels is the same as one of the

17 corresponding blocks of pixels stored in the cache, the

18 caching sub-algorithm does not retransmit this portion of the

19 video image. Instead, a "cache hit" message is sent to the

20 remote participant equipment, which indicates that the most

21 recently transmitted block is already stored in the cache

22 (step **407**). The "cache hit" message contains information

23 regarding which cache contains the corresponding block of

1 pixels, thereby allowing the remote participation equipment to

2 retrieve the block of pixels from its cache and use it do

3 create the video image to be displayed on its attached video

4 display device.

5   The next step in the process, step **409,** determines if the

6 NRDT determined that the block of pixels has changed since the

7 corresponding block of pixels in the compare frame. This step

8 can also be implemented before or in parallel with step **405.**

9 Also, steps **421, 405,** and **407** may be eliminated entirely.

10   The main purpose of step **409** is to determine whether the

11 block has changed since the last frame. If the block has not

12 changed, there is no need to send an updated block to the

13 remote participant equipment. Otherwise, if the block of

14 pixels has changed, it is prepared for compression (step **411**).

15 In the preferred embodiment, step **409** uses a different

16 technique than step **405.** With two ways of checking for

17 redundancy, higher compression will result. Both steps **409**

18 and **411** are executed by a caching sub-algorithm executed by

19 microprocessor **223** of video server **102** (FIG. 2).

20   For any areas of the image that have changed, the cache

21 is updated, and the data is compressed before being sent to

22 the server stack. In the preferred embodiment, the image is

23 compressed using IBM's JBIG compression algorithm. JBIG is

1   designed to compress black and white images. However, the

2   present invention is designed to transmit color video images.

3   Therefore, bit planes of the image are extracted (step **411**),

4   and each bit plane is compressed separately (step **413**).

5   Finally, the compressed image is transmitted to server stack

6   **417** (step **415**), which transmits the data to the remote

7   participant equipment. Server stack **417**, implemented with

8   communication devices **118** and **120** (FIG. 2), enables the

9   compressed video to be sent to remote participant equipment

10   **124** and **138** using any of the various communication mediums or

11   protocols discussed above.

12   Figures 5A and 5B provide detailed flowcharts of a

13   preferred embodiment of the compression process. As seen in

14   FIG. 5A, the video capture is done at a rate of 20 frames per

15   second (step **501**). VGA capture block is implemented by A/D

16   converter **201** (FIG. 2). Standard monitors often update at

17   refresh rates as high as 70 times per second. As a rate of 20

18   frames per second is significantly less frequent, this step

19   limits the amount of data that is captured from the computer.

20   Thus, this first step reduces the bandwidth needed to transmit

21   the video. In this embodiment, the data is outputted in RGB

22   format where 5 bits are allocated to each color. This allows

23   for the representation of 32,768 unique colors. However,

1   other formats capable of storing more or less colors may be

2   used depending on the needs of the users and the total

3   available bandwidth.

4   The digital representation of the captured video image

5   created in step **501** is transferred and stored in either frame

6   buffer 0 **503** or frame buffer 1 **505**. A frame buffer is an area

7   of memory that is capable of storing one frame of video. The

8   use of two frame buffers allows faster capture of image data.

9   The captured frames of video are stored in frame buffer 0 **503**

10 and frame buffer 1 **505** in an alternating manner. This allows

11 the next frame of video to be captured while compression is

12 being performed on the previous frame of video. In video

13 server **102**, frame buffer 0 **503** and frame buffer 1 **505** comprise

14 a portion of frame buffers **227** (FIG. 2).

15   An NRDT test is performed on each block of pixels stored

16 in raw frame buffer 0 **503** and raw frame buffer 1 **505** (step

17 **519**), which compares each block of the captured video image to

18 the corresponding block of the previously captured video

19 image. Step **519** compares blocks of pixels from the video

20 image stored in the current raw frame buffer (i.e., raw frame

21 buffer 0 **503** or raw frame buffer 1 **505**) with the corresponding

22 block of pixels stored in compare frame buffer **521**. This step

1 is discussed in greater detail below with respect to FIGS. 7A

2 and 7B.

3   If step **519** determines that the current block of pixels

4 has changed, then nearest color match function processes the

5 video images contained in raw frame buffer 0 **503** and raw frame

6 buffer 1 **505** (step **509**) in conjunction with the information

7 contained in the client color code table ("CCT from client")

8 **511,** which is stored in flash memory **239** (FIG. 2). The

9 nearest color match function can be executed as software by

10 microprocessor **223**. A detailed explanation of the nearest

11 color match function is provided below with respect to FIG. 6.

12   The CCT obtained from CCT **513** by the nearest color match

13 function is used for color code translation (step **515**), which

14 translates the digital RGB representation of each pixel of the

15 changed block of pixels to reduce the amount of digital data

16 required to represent the video data. Color code translation

17 (step **515**) receives blocks of pixels that the NRDT sub-

18 algorithm (step **519**) has determined have changed relative to

19 the previous captured video image. Color code translation

20 then translates this digital data into a more compact form and

21 stores the result in coded frame buffer **517**. Coded frame

22 buffer **517** can be implemented as a portion of RAM **241** (FIG.

23 2).

1   Alternatively, steps **509** and **515** may be performed in

2   parallel with step **519**. Performing these steps in parallel

3   will reduce the overall amount of processing time required for

4   each block of pixels that has changed. In this scenario,

5   steps **509** and **515** are performed in anticipation of the block

6   of pixels having changed. If this is the case, the processing

7   for steps **509** and **515** may be completed at the same time as the

8   processing for step **519** is completed. Therefore, the

9   algorithm may move directly to step **523** from step **509** without

10 having to wait for the processing of steps **509** and **515**.

11 Otherwise, if step **519** determines that the block of pixels has

12 not changed, and therefore the results of steps **509** and **515**

13 are not required, these results may simply be discarded.

14   Upon completion of step **515**, caching begins by performing

15 a cyclical redundancy check (CRC)(step **523**). Cyclic

16 redundancy check (CRC) is a method known in the art for

17 producing a checksum or hash code of a particular block of

18 data. The CRCs may be computed for two blocks of data and

19 then compared. If the CRCs match, the blocks are the same.

20 Thus, CRCs are commonly used to check for errors. Often, a

21 CRC will be appended to a block of transmitted data so that

22 the receiver can verify that the correct data is received.

23 However, in the present invention, the CRC is used to compare

1   a block of pixels with blocks of pixels stored in a cache.

2   Thus, in step **523**, the CRC is computed for each block of

3   pixels that was determined to have changed by the NRDT sub-

4   algorithm. The array of CRCs is stored in CRC array **525**.

5   Turning next to FIG. 5B, depicted is an overview of the

6   caching and bit splicing/compression sub-algorithms. This

7   portion of the algorithm begins waiting for information from

8   coded frame buffer **517** and CRC array **525** (step **527**). Next, a

9   decision is made as to whether a new video mode has been

10   declared (step **529**). A new video mode can be declared if, for

11   example, a new piece of remote participant equipment is

12   connected to the video transmission system that has different

13   bandwidth or color requirements. If a new video mode has been

14   declared, all data is invalidated (step **531**) and the sub-

15   algorithm returns to step **527** to wait for new information from

16   coded frame buffer **517** and CRC array **525**. Steps **527**, **529**, and

17   **531** are all steps of the overall compression algorithm that is

18   executed by microprocessor **223**.

19   If in step **529** it is deemed that a new video mode has not

20   been declared, then the comparison of the current block of

21   pixel's CRC with the cached CRCs is performed (step **533**).

22   This block compares the CRC data of the current video frame

23   contained in CRC array **525** with the cache of previous CRCs

1 contained in block info array **535**. Block info array **535**

2 stores the cache of pixel blocks and the CRCs of the pixel

3 blocks and can be implemented as a device in RAM **241** (FIG. 2).

4 Step **533** is also a part of the overall compression algorithm

5 executed by microprocessor **223**.

6   Next, a determination is made as to whether the current

7 block of pixels is located within the pixel block cache

8 contained in block info array **535** (step **537**). If it is, a

9 cache hit message is sent to the remote participation

10 equipment and the block of pixels is marked as complete, or

11 processed (step **539**). Since the remote participation

12 equipment contains the same pixel block cache as video server

13 **102** (FIG. 2), the cache hit message simply directs the remote

14 participation equipment to use a specific block of pixels

15 contained in its cache to create the portion of the video

16 image that corresponds to the processed block of pixels.

17   Next, a check is performed for unprocessed blocks of

18 pixels (step **539**). All blocks of pixels that need to be

19 processed, or updated, are combined to create a compute next

20 update rectangle. If there is nothing to update (i.e., if the

21 video has not changed between frames), then the algorithm

22 returns to step **527** (step **543**). Thus, the current frame will

23 not be sent to the remote participation equipment. By

1   eliminating the retransmission of a current frame of video,

2   the sub-algorithm reduces the bandwidth required for

3   transmitting the video.

4   If, however, there are areas of the image that need to be

5   updated, the update rectangle is first compressed. The update

6   rectangle must first be bit sliced (step **545**). A bit plane of

7   the update rectangle is constructed by taking the same bit

8   from each pixel of the update rectangle. Thus, if the update

9   rectangle includes 8-bit pixels, it can be deconstructed into

10   8 bit planes. The resulting bit planes are stored in bit

11   plane buffer **547**. Again, steps **541, 543,** and **545** are all part

12   of the bit splicing/compression sub-algorithm executed by

13   microprocessor **223** of video server **102** (FIG. 2).

14   Each bit plane is compressed separately by the

15   compression sub-algorithm (step **549**). In this case,

16   compression is performed on each bit plane and the resulting

17   data is sent to server stack **417** (step **551**). In the preferred

18   embodiment, compression is performed by JBIG **229** (FIG. 2)(step

19   **549**). Thereafter, the compressed bit planes are sent to the

20   remote participant equipment via communication device **118** or

21   **120**.

22   Since the preferred embodiment captures frames 20 times

23   per second, it is necessary to wait 300 ms between video frame

1   captures. Thus, the algorithm will wait until 300 ms have

2   passed since the previous frame capture before returning the

3   sub-algorithm to step **527** (step **553**).

4   Referring now to FIG. 6, illustrated is the nearest color

5   match function (step **509** of FIG. 5A) that selectively converts

6   less frequently occurring colors to more frequently occurring

7   colors by mapping the less frequently occurring colors to the

8   more frequently occurring colors via use of a CCT. Nearest

9   color match function **509** processes one block of pixels of the

10   video image stored in raw frame buffer 0 **503** or raw frame

11   buffer 1 **505** at a time. As shown in FIG. 6, a block of pixels

12   is extracted from the video image stored in raw frame buffer 0

13   **503** or raw frame buffer 1 **505** (step **600**). In the preferred

14   embodiment, the extracted block has a size of 64 by 32 pixels.

15   However, nearest color match function **509** may process blocks

16   having any size.

17   The goal of the nearest color match function is to

18   eliminate noise introduced by the A/D conversion. This is

19   accomplished by converting less frequently occurring pixel

20   values to similar, more frequently occurring pixel values.

21   This is done primarily via histogram analysis and difference

22   calculations. Nearest color match function **509** generates a

23   histogram of pixel values (step **601**). The histogram measures

1 the frequency of each pixel value in the block of pixels

2 extracted during step **600**. The histogram is sorted, such that

3 a list of frequently occurring colors (popular color list **603**)

4 and a list of least frequently occurring colors (rare color

5 list **605**) are generated. The threshold for each list is

6 adjustable.

7   Nearest color match function **509** analyzes each low

8 frequently occurring pixel to determine if the pixel should be

9 mapped to a value that occurs often. First, a pixel value is

10 chosen from rare color list **605** (step **607**). Then, a pixel

11 value is chosen from popular color list **603** (step **609**). These

12 distance between these two values is then computed (step **611**).

13 In this process, distance is a metric computed by comparing

14 the separate red, green and blue values of the two pixels.

15 The distance value, D, can be computed in a variety of ways.

16 One such example is:

17 $$D = (R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2$$

18   In this formula, R1 is the red value of the low frequency

19 pixel, R2 is the red value of the high frequency pixel, G1 is

20 the green value of the low frequency pixel, G2 is the green

21 value of the high frequency pixel, B1 is the blue value of the

22 low frequency pixel, and B2 is the blue value of the high

23 frequency pixel.

1   This formula yields a distance value, D, which indicates

2 the magnitude of the similarity or difference of the colors of

3 two pixels, such as a less frequently occurring pixel versus a

4 more frequently occurring pixel. The goal of the sub-

5 algorithm is to find a more frequently occurring pixel having

6 a color that yields the lowest distance value when compared to

7 the color of a less frequently occurring pixel. Therefore, a

8 comparison is performed for each computed distance value (step

9 **613**). Every time a distance value is computed that is less

10 than all previous distance values, the distance value is

11 written to the closest distance variable (step **615**).

12   Once it is determined that all more frequently occurring

13 pixels have been compared to less frequently occurring pixels

14 (step **617**), a computation is performed to see if the lowest

15 occurring D is within a predefined threshold (step **619**). If

16 this D is within the threshold, CCT **513** is updated by mapping

17 the low frequently occurring pixel to the color code value of

18 the high frequently occurring pixel that yielded this D value

19 (step **621**). This process is repeated for all low frequency

20 pixels and CCT **513** is updated accordingly.

21   Turning to FIG. 7A, RGB NRDT step **519** (FIG. 5A) is

22 illustrated in further detail. This process operates on every

23 block of pixels. Current pixel block **700** represents a block

1 of pixels of the video image contained in the current raw

2 frame buffer (i.e., raw frame buffer 0 **503** or raw frame buffer

3 1 **505** (FIG. 5A)). Previous pixel block **701** contains the

4 corresponding block of pixels of the video image contained in

5 compare frame buffer **521** (FIG. 5A). Step **519** begins by

6 extracting corresponding pixel values for one pixel from the

7 current pixel block **700** and previous pixel block **701** (step

8 **703**). Then, the pixel color values are used to calculate a

9 distance value, which indicates the magnitude of the

10 similarity or difference between the colors of the two pixels

11 (step **705**). In the preferred embodiment of the present

12 invention, the distance value is computed using the following

13 formula:

14
$$D = (R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2$$

15 As before, R1, G1, and B1 are the red, green and blue

16 values respectively of the frame buffer pixel. Similarly, R2,

17 G2, and B2 are the red, green and blue values respectively for

18 the compare frame buffer pixel.

19 Next, the computed distance value D is compared with a

20 pixel threshold (step **707**). If D is greater than the pixel

21 threshold, it is added to an accumulating distance sum (step

22 **709**). If the value of D is less than the pixel threshold, the

1 difference is considered to be insignificant or noise, and it

2 is not added to the distance sum.

3 This process of computing distance values and summing

4 distance values that are greater than a predefined pixel

5 threshold continues until it is determined that the last pixel

6 of the block of pixels has been processed (step **711**). Once

7 the last pixel is reached, the distance sum is compared with a

8 second threshold, the block threshold (step **713**). If the

9 distance sum is greater than the block threshold, the current

10 block of pixels designated as changed as compared to the

11 corresponding block of pixels from the previously captured

12 frame. Otherwise, if the distance sum is less than the block

13 threshold, the block of pixels is designated as unchanged.

14 If the block of pixels is designated as changed, step **715**

15 is executed. Step **715** sets a flag that indicates that the

16 particular block of pixels has changed. This flag is

17 transmitted to all of the remote participant equipment.

18 Furthermore, the new block of pixels is written to compare

19 frame buffer **521** (FIG. 5A) to replace the corresponding

20 previous block of pixels.

21 Otherwise, if the distance sum does not exceed the block

22 threshold, the block is designated unchanged and, a flag is

23 set to indicate that this block of pixels does not need to be

1   re-transmitted to the remote participation equipment (step

2   **721**). Rather, the remote participation equipment will

3   recreate the portion of the video image represented by the

4   block of pixels using the same block of pixels displayed for

5   the previous frame of video. At this point, the system

6   computes CRCs for changed blocks of pixels (step **523** of FIG.

7   5A), as discussed in greater detail above with respect to FIG.

8   5A.

9   Figure 7B further illustrates the two level thresholding

10 used by the NRDT sub-algorithm shown in FIG. 7A. For

11 illustration purposes only, 4x4 blocks of pixels are shown.

12 Each pixel is given red, green, and blue color values that

13 range from 0 to 255, as is commonly performed in the art. A

14 pixel having red, green, and blue values of 0 represents a

15 black pixel, whereas a pixel having red, green, and blue

16 values of 255 represents a white pixel. Previous pixel block

17 **751** is a block of pixels grabbed from compare frame buffer **521**

18 (FIG. 5A). Previous pixel 1 **752** is the pixel in the upper,

19 left corner of previous pixel block **751**. Since every pixel of

20 previous pixel block **751** has a value of 0, previous pixel

21 block **751** represents a 4x4 pixel area that is completely

22 black.

1   Current pixel block **753** represents the same spatial area

2   of the video frame as previous pixel block **751**, but it is one

3   frame later. Here current pixel 1 **754** is the same pixel 1 as

4   previous pixel 1 **752**, but is one frame later. For simplicity,

5   suppose a small white object, such as a white cursor, enters

6   the area of the video image represented by previous pixel

7   block **751**. This change occurs in current pixel 1 **754** of

8   current pixel block **753**. In current pixel block **753**, the

9   majority of the pixels remained black, but current pixel 1 **754**

10 is now white, as represented by the RGB color values of 255,

11 255, and 255.

12 Further suppose that noise has been introduced by the A/D

13 conversion, such that previous pixel **755** has changed from

14 black, as represented by its RGB values of 0, 0, and 0, to

15 gray. The new gray color is represented by the RGB values of

16 2, 2, and 2 assigned to current pixel **756**.

17 Further suppose that the pixel threshold is 100, and the

18 block threshold is 200. The NRDT sub-algorithm calculates the

19 distance value between each pixel of current pixel block **753**

20 and previous pixel block **751**. The formula used in the

21 preferred embodiment of the present invention, as discussed

22 above with respect to FIG. 7A, is:

23 $$D = (R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2$$

1   Therefore, the distance value between current pixel 1 **754** and

2   previous pixel 1 **752** is:

3                  $D = (255 - 0)^2 + (255 - 0)^2 + (255 - 0)^2$

4   or 195,075. This distance value is added to the distance sum

5   because 195,075 exceeds the pixel threshold of 100. However,

6   the distance value between the black previous pixel **755** and

7   the gray current pixel **756** is not added to the distance sum

8   because the distance between the pixels, as calculated using

9   the above distance formula, equals 12, which does not exceed

10   the pixel threshold of 100. Similarly, the distance value is

11   computed for all of the remaining pixels in the two pixel

12   blocks. Each of these distance values equals zero, therefore,

13   since these distance values are less than the pixel threshold,

14   they are not added to the distance sum.

15   Consequently, after the distance values for all pixels

16   have been processed, the distance sum equals 195,075. Since

17   this value is greater than the block threshold of 200, the

18   block is designated. This example illustrates the advantages

19   of the two-level thresholding feature of the NRDT sub-

20   algorithm. That is, the noise that occurred in current pixel

21   **756** of current pixel block **753** was ignored, whereas the real

22   change in video that occurred in current pixel 1 **754** of

23   current pixel block **753** was recognized.

1   Turning finally to FIG. 8, shown is a flowchart of the

2   decompression algorithm executed by remote participant

3   equipment (e.g., video client **124** or remote computer **138** (FIG.

4   1)). The decompression algorithm begins by waiting for a

5   message (step **801**). This message is transmitted from server

6   stack **417** of video server **102** via communication medium **114** or

7   **116** to communication device **122** or **124**, respectively (FIG. 1).

8   Thereafter, microprocessor **333** or **303** receives the information

9   from communication device **122** or **124** (FIGS. 3A and 3B),

10   respectively, and writes the data to client stack **803**. Client

11   stack **803** may be a register in microprocessor **333** or **303**,

12   memory **331** or **307**, or some other device capable of permanently

13   or temporarily storing digital data. In one embodiment of the

14   present invention, messages are transmitted using a TCP/IP

15   communication protocol. In this scenario, client stack **303** is

16   the local TCP/IP stack. Other embodiments may use a protocol

17   other than TCP/IP. However, irrespective of the communication

18   protocol, the present invention uses client stack **303** to store

19   received messages for processing.

20   Once a message is received in client stack **303**, it is

21   processed to determine whether the message is a new video mode

22   message (step **805**). A new video mode message may be sent for

23   a variety of reasons including a bandwidth change, a change in

1   screen resolution or color depth, a new client, etc. This

2   list is not intended to limit the reasons for sending a new

3   video mode message, but instead to give examples of when it

4   may occur. If the message is a new video mode message,

5   application layer **823** is notified of the new video mode (step

6   **807**). According to the preferred embodiment, application

7   layer **823** is software executed by microprocessor **303** or **333**

8   that interfaces with the input and output devices of the

9   remote participation equipment (e.g., video display device **126**

10 or **146**, etc.). Any video updates must therefore be sent to

11 application layer **823**. Also, the old buffers are freed,

12 including all memory devoted to storing previously transmitted

13 frames, and new buffers are allocated (step **809**). The

14 decompression algorithm then returns to step **801**.

15   If the new message is not a video mode message, the

16 message is further processed to determine if it is a cache hit

17 message (step **811**). If yes, the cache hit message is

18 deciphered to determine which block of pixels, of the blocks

19 of pixels stored in the three cache frame buffers **815**, should

20 be used to reconstruct the respective portion of the video

21 image. Although three cache frame buffers **815** are used in the

22 preferred embodiment of the present invention, any quantity of

23 cache frame buffers may be used without departing from the

1   spirit of the invention.   Cache frame buffers **815** store the

2   same blocks of pixels that are stored in the cache frame

3   buffers located internal to video server **102** (FIG. 2).   Thus,

4   the cache hit message does not include video data, but rather

5   it simply directs the remote participation equipment as to

6   which block of pixels contained in the cache frame buffer **815**

7   should be sent to merge frame buffer **817**.   The block of pixels

8   contained within the specified cache is then copied from cache

9   frame buffer **815** to merge buffer **817** (step **813**).   Finally,

10   application layer **823** is notified that an area of the video

11   image has been updated (step **825**).   Merge buffer **817** contains

12   the current representation of the entire frame of video in

13   color code pixels.   Application layer **823** copies the pixel

14   data from merge buffer **817** and formats the data to match the

15   pixel format of the connected video display device **126** or **146**

16   (step **819**).   Thereafter, the formatted pixel data is written

17   to update frame buffer **821**, which then transmits the data to

18   video display device **126** or **146**.   Alternatively, in lieu of a

19   video display device, the formatted pixel data may be written

20   to a video card, memory, and/or any other hardware or software

21   commonly used with video display devices.

22   Further, if the new message is not a new video mode or

23   cache hit message, it is tested to determine if it is a

1   message containing compressed video data (step **827**). If the

2   message does not contain compressed video data, the

3   decompression algorithm returns to step **801** and waits for a

4   new message to be transmitted from server stack **417**.

5   Otherwise, if the message does contain compressed video data,

6   the data is decompressed and transferred to bit plane frame

7   buffer **833** (step **829**). As described above, the preferred

8   embodiment incorporates the JBIG lossless compression

9   technique. Therefore, decompression of the video data must be

10 performed for each individual bit plane. After each bit plane

11 is decompressed, it is merged with previously decompressed bit

12 planes, which are stored in bit plane frame buffer **833** (step

13 **829**). When a sufficient number of bit planes have been

14 merged, the merged data contained in bit plane frame buffer

15 **833** is transferred to merge frame buffer **817** (step **831**).

16 Alternatively, individual bit planes may be decompressed and

17 stored directly in merge frame buffer **817**, thereby eliminating

18 step **831**. When all of the data required to display a full

19 frame of video is transferred to merge frame buffer **817**,

20 application layer **823** copies the data in merge frame buffer

21 **817** to update frame buffer **821** (step **819**). Thereafter, the

22 data is transferred to video display device **126** or **146**.

1   In an alternate embodiment, the video displayed on video

2   display device **126** or **146** can be updated after each bit plane

3   is received. In other words, a user does not have to wait

4   until the whole updated frame of video is received to update

5   portions of the displayed video. This alternative method is

6   desirable when the bandwidth available for video transmission

7   varies. Also, this progressive method of updating the video

8   display is one of the advantages of using the JBIG compression

9   algorithm.

10   Next, the decompression algorithm determines whether all

11   of the color code data from one field of the current video

12   frame has been received (step **835**). If a full field has not

13   been received, the decompression algorithm returns to step **801**

14   and waits for the remainder of the message, which is

15   transmitted from server stack **417** to client stack **803** in the

16   form of a new message. Otherwise, if a full field has been

17   received, the decompression method notifies application layer

18   **823** (step **837**). Similar to that described above with respect

19   to processing cache hit messages, this notification directs

20   application layer **823** to read the data in merge frame buffer

21   **817** and convert it to the current screen pixel format (step

22   **819**). Thereafter, the formatted data is written to update

1   frame buffer **821**, which transmits the data to video display

2   device **126** or **146**.

3   After a full field has been received and application

4   layer **823** has been notified, a second determination is made to

5   determine if the full field is the last field included in the

6   message. If it is, the newly decompressed block of pixels is

7   written to one of the cache frame buffers **815** (step **841**).

8   Otherwise, the decompression algorithm returns to step **801** and

9   continues to wait for a new message. Preferably, the new

10 block of pixels written to cache frame buffer **815** overwrites

11 the oldest block of pixels contained therein. Step **841**

12 ensures that the cache is up-to-date and synchronized with the

13 cache of video server **102**. After the completion of the cache

14 update, the decompression algorithm returns to step **801**.

15   While the present invention has been described with

16 reference to one or more preferred embodiments, which

17 embodiments have been set forth in considerable detail for the

18 purposes of making a complete disclosure of the invention,

19 such embodiments are merely exemplary and are not intended to

20 be limiting or represent an exhaustive enumeration of all

21 aspects of the invention. The scope of the invention,

22 therefore, shall be defined solely by the following claims.

23 Further, it will be apparent to those of skill in the art that

1   numerous changes may be made in such details without departing

2   from the spirit and the principles of the invention.